Greg Chan

EECS 349 Extra Credit Homework

Design Choices

Part A

I decided to build my dictionary by excluding the header information. I did this by searching for the subject line and using that to find the body of the message. I then built my dictionary by splitting each line in the email by spaces. I chose to include HTML tags and links because I think it is a somewhat fair assumption that most emails you receive do not contain html tags. Additionally, I felt that I wanted to include punctuation in words because sentences that end with certain words might only appear in spam emails. I also made the decision to remove words that contain no characters because these could be line breaks are are likely to be equally probable in both spam and ham. I did not limit the dictionary size because I feel like that would only reduce the chance of spam or ham classification. Additionally, I also added a pseudo count to the probabilities so that the probability of a word not appearing in either spam or ham is never zero.

Part B

I decided to design my spam sorter by taking all the unique words in a message and getting their probabilities as listed in the spam and ham dictionaries. If the word was not in the spam dictionary or the word was not in the ham dictionary, then it would not be counted in the probability calculation. Since the probabilities from the dictionary were low, below one percent, I decided to use the log formulation so that I could get more accurate spam and ham classification. Equation 7 and equation 6 are equivalent because of the rule of logs, which says that if you have a much of things multiplied inside a log function, you can split this into a sum of logs. So equation 7 is simply equation 6 but with a log around all of equation 6, which is then split up using the rule of logs.

Datasets:

I decided to use all of the spam and all of the ham to build my dictionary. I also decided to test on all the spam and ham to see what impact this might have on classification. While I understand that building the dictionary using all the spam and ham might result in a bias towards correct classification, I think that it is also important to note that this is mostly probability based and my dictionary was built using just the body of the emails so in this way it might reduce the bias.

Testing

For testing purposes, I trained on all the spam and all the ham to build my dictionary. Then I put 4 random emails for the spam and 4 random emails from the ham and put them in the mail_directory. I then read in the dictionary that I trained on earlier and ran it on these messages. I was then able to determine if the spam was classified correctly by comparing the file name to the file names in the spam training directory.

As a final test, I also ran this on all of the emails. To determine if the classification worked on a larger scale. I used the same method to determine if the classification was correct by comparing the file names to the file names in the spam training directory or ham training directory. Since the files were taken from these two sets, if the file existed in the spam directory then it was spam and if it existed in the ham directory it was ham.

The flaw in the testing and likely induced bias is that I chose a subset or the same set as the training set to classify and test. Given more time, I would have tested it on a larger training set of unseen emails.

Future Work:
I would try to redesign the dictionary to better encompass a list of actual words as opposed to potential words, and by this I mean I would try to separate by not only spaces, but also try to pick out actual english words. I don't think I would exclude certain portions of the emails, but I might include more about the header information. I found it difficult to parse the header information and thus decided that the repeated words in these sections might through off the probability that a document is spam or ham. I think these headers do contain important information like the source of the email. Additionally, I think I would also include the subject line and recipients because this might be indicative of spam or ham. All of this could be done through more experimentation. I think to parse the header and subject information, I would need a more sophisticated parser that could identify information after these headings, and identify when these headings end. I'm not sure specifically how this might work, but one somewhat feasible way to do this would to split lines by colons and take only what is after a colon, but only do this for the part of the email that is above the message. There are many improvements that can be made to this Niave Bayes spam filter, but I think improving the dictionary by including more information about the email could definitely help.